

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-69222

(43) 公開日 平成10年(1998) 3月10日

(51) Int. Cl. ⁸	識別記号	片内整理番号	F I	技術表示箇所
G 0 9 C 1/00	6 5 0	7259-5 J	G 0 9 C 1/00	6 5 0 A
	6 3 0	7259-5 J		6 3 0 Z
	6 6 0	7259-5 J		6 6 0 A
		7259-5 J		6 6 0 G
G 0 6 K 17/00			G 0 6 K 17/00	S

審査請求 未請求 請求項の数 4 O L (全 9 頁) 最終頁に続く

(21) 出願番号 特願平8-224971

(22) 出願日 平成8年(1996) 8月27日

(71) 出願人 000002897

大日本印刷株式会社

東京都新宿区市谷加賀町一丁目1番1号

(72) 発明者 入澤 和義

東京都新宿区市谷加賀町一丁目1番1号

大日本印刷株式会社内

(72) 発明者 半田 富己男

東京都新宿区市谷加賀町一丁目1番1号

大日本印刷株式会社内

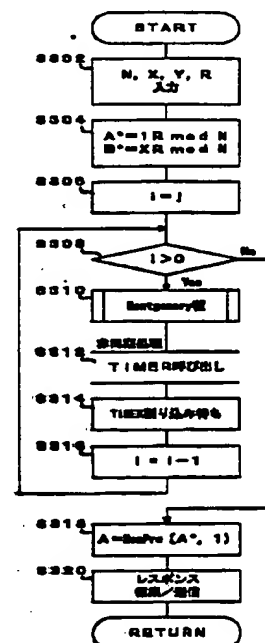
(74) 代理人 弁理士 鎌田 久男

(54) 【発明の名称】 I Cカード

(57) 【要約】

【課題】 コマンドに対する応答情報の送信時より暗号化処理又は復号化処理に用いた鍵情報の内容を推測されることがない I Cカードを提供する。

【解決手段】 CPUと、CPUがアクセス可能なメモリとを備え、外部命令に従ってメモリに保存されている鍵情報を用いてデータの暗号化処理又は復号化処理を行い、その結果に関する応答情報を外部へ送信する I Cカードにおいて、暗号化処理又は復号化処理の実行中又は実行の前後に、鍵情報の内容との相関関係を喪失させるように、応答情報の送信時を遅延 (S 3 1 4 ~ S 3 1 6) させる遅延処理を実行する。



【特許請求の範囲】

【請求項1】 CPUと、

前記CPUがアクセス可能なメモリとを備え、
外部命令に従って前記メモリに保存されている鍵情報を用いてデータの暗号化処理又は復号化処理を行い、その結果に関する応答情報を外部へ送信するICカードにおいて、

前記応答情報の送信時を遅延させる遅延手段を有し、
前記暗号化処理又は復号化処理の実行中又は実行の前後に前記遅延手段を実行することにより、前記鍵情報の内容と、前記応答情報の送信時との相関関係を喪失させることを特徴とするICカード。

【請求項2】 請求項1に記載のICカードにおいて、
前記遅延手段は、前記CPUが無作為な時間実行する、
前記暗号化処理又は復号化処理と実質的に無関係な演算処理であることを特徴とするICカード。

【請求項3】 請求項1に記載のICカードにおいて、
所定時間の経過を通知する計時手段を有し、
前記遅延手段は、前記CPUが前記計時手段から前記通知があるまで前記暗号化処理又は復号化処理の開始又は続行を中断することであることを特徴とするICカード。

【請求項4】 請求項1から請求項3までのいずれか1項に記載のICカードにおいて、
前記遅延手段は、前記鍵情報のビット構成によらず、前記暗号化処理又は復号化処理に要する時間を一定とすることを特徴とするICカード。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、外部からの命令に従いデータを暗号化又は復号化するICカードに関するものである。

【0002】

【従来の技術】ICカードは、磁気カードに代わる新しい情報記憶媒体として、近年注目を集めている。特に、CPUを内蔵したICカードは、単なる情報記憶媒体としての機能だけでなく、情報処理機能をも有し、高度なセキュリティを実現できることから、高度情報化社会の種々の分野における利用が期待されている。一般にICカードでは、EEPROMなどの不揮発性メモリが内蔵されており、このEEPROM内にファイルとして情報が記憶される。EEPROMへのアクセスは、ICカードに外部よりコマンドを与え、そのコマンドを内蔵のCPUが解釈・実行することにより行われる。各ファイルには、予め所定のアクセス条件が設定されており、CPUは、コマンドの引数がアクセス条件を満足している場合に限りファイルへアクセスする。これによりICカードは、正当な権限を有さない第三者が不正にEEPROMのデータを改竄する又は盗むことを防止している。

【0003】さらに、ICカードシステムでは、ICカ

ードとリーダ・ライタとの間でデータの送受信を行う場合は、そのデータを暗号化する。これは、リーダ・ライタとICカードとの間の通信信号を第三者が不正に取得した場合においても、その内容が盗まれることを防止するためである。図8は、リーダ・ライタ等の外部機器からICカードへの情報の伝達を示す図である。外部機器は、平文Aに所定の暗号化処理を加えて暗号文Xを取得し、これをICカードに送信する。ICカードでは、外部機器が用いた暗号化処理に対応する復号化処理を行うことにより暗号文Xを復号化する。復号化処理が終了すると、ICカードは、処理が終了した旨のレスポンス情報を編集し、これを外部機器に送信する。また、外部機器は、レスポンス情報を受信して一連の処理を終了する。

【0004】平文の暗号化は、通常、平文と所定の鍵情報とを変数とする数式を計算することにより実行される。このような暗号化手法として、現在提案されている代表的なものに、例えばR. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems" (Communications of the ACM, Vol. 21, No. 2: pp. 120-126, Feb., 1978) に論及されているRSA暗号がある。RSA暗号は、平文を暗号化する際に使用する鍵情報と、暗号文を復号化する際に使用する鍵情報とが異なる、いわゆる非対称鍵方式の暗号化方法である。非対称な2種類の鍵情報のうち、一方はICカードのEEPROM等に第三者に対し秘密に格納される秘密鍵として、他方は第三者に広く公表される公開鍵として利用される。

【0005】RSA暗号の復号化処理は、式「 $A = X^Y \pmod{N}$ 」に従い実行される。ここでAは平文、Xは暗号文、Yは秘密鍵、Nは公開鍵である。上記式から明らかなように、RSA暗号では、べき乗剰余の計算を行うことにより復号化処理が行われる。したがって、その計算量は通常非常に大きい。このために、RSA暗号を実行する場合には、通常、べき乗剰余の計算を軽減するための計算アルゴリズムが利用される。べき乗剰余の計算量を軽減する計算アルゴリズムとしては、例えば「べき乗の2進計算法」が知られている(D. E. Knuth, The Art of Computer Programming, volume 2, Semi-numerical Algorithms, Addison-Wesley, 2nd edition, 1981, 参照)。

【0006】RSA暗号の計算には、この「べきの2進計算法」を応用し、さらに剰余演算を少ない計算量で行うことを可能としたMontgomery法といわれる計算アルゴリズムが利用される(P. L. Montgo

mery. "Modular Multiplication without Trial Division", Mathematics of Computation, Vol. 44, No. 170, pp. 519-521, Apr., 1985, 参照)。Montgomery法では、図9に示すアルゴリズムにより、べき乗剰余の計算「 $A=X^Y \pmod{N}$ 」を実行する。

【0007】まずN、X、Y及びRの4つの値が入力される(S902)。ここでは、Xは、Nに対し $0 < X < N$ の関係を満たす。また、秘密鍵Yは、2進法表示により $Y=e_je_{j-1}\cdots e_2e_1$ と表現されるものとする。ただし、 e_i は第iビット目の数値を意味する。Rは、Yのビット数jを用いて $R=2^j$ と定義される数値である。次に、入力された数値よりMontgomery剰余 A^* 、 B^* が求められる(S904)。Montgomery剰余とは、nを法とする剰余「 $a \pmod{n}$ 」と一対一に対応するように「 $a^*=ar \pmod{n}$ 」と定義される値である。ただし、nはk-bitsの整数であり、 $2^{k-1} \leq n < 2^k$ 、 $r=2^k$ 、 $\gcd(r, n)=1$ 、である。

【0008】次に指数Yの各ビットについてMontgomery積の計算が行われる(S906~S912)。つまり、指数Yがjビットから構成されている場合には、S910のMontgomery積の計算がj回反復して実行される。なお、Montgomery積とは、「 $\text{MonPro}(a^*, b^*) = a^* b^* r^{-1} \pmod{n}$ 」と定義される積である。図10は、Montgomery積の処理内容を示す流れ図である。Montgomery積には、S1002及びS1006の2つの演算処理が含まれてい。S1002の演算処理は、指数Yの各ビットについて、それが1であるか0であるかに関わらず必ず行われる。一方、S1006の演算処理は、指数Yを構成するビットのうち1であるものについてのみ実行される。図9におけるMontgomery積の反復計算(S906~S912)が指数Yを構成する全てのビットについて終了すると、次に A^* と1についてのMontgomery積が実行される(S914)。この結果、暗号文Xを復号化した平文Aが取得され、一連の復号化処理が終了する。

【0009】

【発明が解決しようとする課題】上記に説明したように、従来のICカードの復号化処理では、2進法表示された指数Yの各ビット桁に対して、1が立っているものと、1が立っていないものとの処理の内容が異なっている。すなわち、図10における処理において、該当ビットが1ならばS1002及びS1006の2ステップ、該当ビットが0ならばS1002の1ステップのみの計算が行われる。したがって、復号化処理に要する時間は、べき乗の指数に占める1のビット数に依存し、1のビット数が多いほど長くなる。

【0010】このために、第三者は、ICカードに暗号文の復号化を命じるコマンドを送信してからレスポンスが返信されるまでの時間より、ICカードが復号化処理に必要なとした時間を求め、さらに求めた時間よりべき乗剰余計算の指数(秘密鍵)Yにおける1のビットと0のビットの割合を推定することが可能性であった。つまり、従来のICカードでは、レスポンスの送信時から秘密鍵Yの内容が解読され、ICカードのセキュリティが害される恐れがあるという問題があった。

10 【0011】そこで、本発明の課題は、外部からのコマンドに対する応答情報の送信時より暗号化処理又は復号化処理に用いた鍵情報の内容を推測されることのないICカードを提供することである。

【0012】

15 【課題を解決するための手段】前記課題を解決するために、請求項1に係る発明は、CPUと、前記CPUがアクセス可能なメモリとを備え、外部命令に従って前記メモリに保存されている鍵情報を用いてデータの暗号化処理又は復号化処理を行い、その結果に関する応答情報を外部へ送信するICカードにおいて、前記応答情報の送信時を遅延させる遅延手段を有し、前記暗号化処理又は復号化処理の実行中又は実行の前後に前記遅延手段を実行することにより、前記鍵情報の内容と、前記応答情報の送信時との相関関係を喪失させることを特徴とする。

25 【0013】請求項2に係る発明は、請求項1に記載のICカードにおいて、前記遅延手段は、前記CPUが無作為な時間実行する、前記暗号化処理又は復号化処理と実質的に無関係な演算処理であることを特徴とする。なお、「実質的に無関係な処理」とは、その処理を行わなくとも暗号化処理又は復号化処理を正常に実行することが可能な処理をいう。また、「無作為な時間実行する」とは、一定の時間を要する処理を無作為の回数実行すること、又は実行のたびに実行時間が無作為に決定される処理を1回又は2回以上実行すること等をいう。

35 【0014】請求項3に係る発明は、請求項1に記載のICカードにおいて、所定時間の経過を通知する計時手段を有し、前記遅延手段は、前記CPUが前記計時手段から前記通知があるまで前記暗号化処理又は復号化処理の開始又は続行を中断することであることを特徴とする。請求項4に係る発明は、請求項1から請求項3までのいずれか1項に記載のICカードにおいて、前記遅延手段は、前記鍵情報のビット構成によらず、前記暗号化処理又は復号化処理に要する時間を一定とすることを特徴とする。

45 【0015】

【発明の実施の形態】以下、図面等を参照して、本発明に係る実施形態について、さらに詳しく説明する。

(第1実施形態) 本発明に係る第1実施形態は、非対称鍵方式の暗号化方法であるRSAを利用して平文の暗号化、又は暗号文の復号化を行うことが可能なICカード

である。本実施形態において、RSAによる暗号化処理又は復号化処理は、図9に示したMontgomery法による計算アルゴリズムに従い行われる。図1は、本実施形態に係るICカードの構成を示す図である。図1に示されるように、ICカード10は、読み出し専用メモリであるROM12、揮発性メモリであるRAM14、随時書換え可能な不揮発性メモリであるEEPROM16、これらのメモリにアクセスするCPU18、及びタイマーモジュール20を備えている。なお、タイマーモジュールとは、CPU18の動作とは独立に動作し、指示された時間が経過するとCPU18に対して割り込みをかけて通知する計時装置である。

【0016】また、ICカード10は、リーダー・ライター（不図示）と電気信号等の授受を行うためのI/Oラインを備えている。ICカードをリーダー・ライターに挿入すると、リーダー・ライターの接点がこのI/Oラインと接続され、電気信号の授受が行われる。CPU18は、上記I/Oラインを介してコマンドを付与される。コマンドとは、リーダー/ライターからICカードへ送られる情報であって、ICカードに所定の動作させるためのものをいう。CPU18は、コマンドを付与されると、ROM12又はEEPROM16に格納されているプログラムを実行することによりそのコマンドを処理する。

【0017】図2は、本実施形態で使用されるコマンドの一つであるRSA_CALCコマンドのフォーマットを示す図である。RSA_CALCコマンドは、ICカード10に暗号文を復号化させ、その結果取得された平文をレスポンスとして返信させるコマンドである。RSA_CALCコマンドの最初の5バイトは、それぞれコマンドのクラスを示すCLA、種別を示すINS、コマンドのパラメータP1、P2、及び後に続くDATAの長さ（バイト数）を示すLCである。第6バイト目以降のDATAは、復号化されるべき暗号文Xである。また、DATAの後に続く1バイトのデータLEは、レスポンスの期待値である。本実施形態では、暗号化又は復号化されたデータを最大256長バイトを限度として全てレスポンスとして返信するよう、LEの値を設定している。

【0018】図3は、RSA_CALCコマンドを実行するときのICカード10の動作を示す流れ図である。RSA_CALCコマンドが受信されると、まず、RSA_CALCのDATAより暗号文Xが取得され、また、EEPROMの所定アドレスに保存されている公開鍵N、秘密鍵Y、及び定数Rが読み出される（S302）。次に、取得された上記数値よりMontgomery剰余A*及びB*が算出される（S304）。さらに、カウンタ*i*が2進表示された秘密鍵Yの桁数（ビット数）*j*にセットされる（S306）。次に、カウンタ*i*が0より大であるかが判断される（S308）。

【0019】S308の判断の結果、カウンタ*i*が0

より大であると、秘密鍵Yを構成するビットのうち、まだMontgomery積の計算を行っていないビットが存在することが意味される。この場合にCPU18は、そのときにカウンタ*i*が示す桁のビットについてMontgomery積の計算を実行する（S310）。S310では、図9のS910で説明したのと同じの処理が行われる。また、CPU18は、S310の処理を開始すると同時に、所定時間を指定してタイマーモジュール20を呼び出す（S312）。ここで所定時間とは、1が立っているビットについてS310のMontgomery積を実行するのに要する時間と等しい時間又はこれを越える時間をいう。これにより、CPU18がS310の処理を実行すると平行して、タイマーモジュール20が所定時間のカウントを行う。

【0020】CPU18は、S310における処理を終了した後は、タイマーモジュール20からの割り込みがあるまで待機する（S314）。割り込みがあると、CPU18は、カウンタ*i*を1だけデクリメントし（S316）、その後S308からS316までの処理を繰り返す。一方S308において、カウンタ*i*が「*i*>0」という条件を満たさない場合には、秘密鍵Yを構成するビットの全てについてMontgomery積の計算がなされたことが意味される。この場合にCPU18は、次にS318に示すA*と1についてのMontgomery積の計算を実行し、平文Aを取得する。さらに、CPU18は、RSA_CALCコマンドが正常に処理された旨のレスポンス情報を編集し、これをリーダー・ライターに送信する（S320）。

【0021】以上説明したように、本実施形態のICカードでは、暗号文の復号化処理を行うべく、S310のMontgomery積を実行する場合には、同時にタイマーモジュールを呼び出すことにより一定の時間を計測し、その一定時間が経過するまでは、たとえMontgomery積の計算が終了している場合であっても、次の処理を実行しないこととしている。これにより、本実施形態では、復号化処理の計算時間は、秘密鍵Yのビット構成によらず常に一定となり、また、ICカード10がRSA_CALCコマンドを受信してからレスポンスを返信するまでの時間も秘密鍵の内容によらず一定となる。よって、本実施形態では、レスポンスが送信される時より、秘密鍵のビット構成が推定されることはなく、極めてセキュリティーの高いICカードを提供することが可能となっている。

【0022】（第2実施形態）次に、本発明に係る第2実施形態について説明する。なお、以下の説明において、第1実施形態と同様な機能を果たす部分には、同一の符号を付し、重複する説明を適宜省略する。図4は、本実施形態のICカード30の構成を示す図である。ICカード30は、タイマーモジュール20を有せず、コプロセッサ32を有し、Montgomery積の計

算をこのコプロセッサ32に実行させる点において第1実施形態のICカード10と異なっている。また、ICカード30は、ICカード10と同様にリーダー・ライターからRSA_CALCコマンドを送信されることにより暗号文Xを復号化し、コマンド処理が終了するとその旨のレスポンスをリーダー・ライターに送信する。復号化処理は、第1実施形態と同様にRSA暗号に基づき行われ、RSA暗号の計算はMontgomery法のアルゴリズムに従い実行される。

【0023】図5は、RSA_CALCコマンドを実行するときのICカード30の動作を示す流れ図である。図5中、S502からS508までは、図3のS302からS308までの内容と同一であり、ICカード30とICカード10との間に動作の相違はない。S508においてカウンタ*i*が「 $i > 0$ 」の条件を満たすと、CPU18は、コプロセッサ32にA*、B*、N及びRの値と、秘密鍵Yの該当ビットの値とを引き渡す。A*等の値の引渡しを受けたコプロセッサ32は、図10に示したMontgomery積の計算を実行する。本実施形態は、このようにMontgomery積の計算をコプロセッサに実行させることにより、復号化処理をより高速に行うことを可能としている。

【0024】一方、コプロセッサ32がMontgomery積の計算を行っている間に、CPU18は、復号化処理とは無関係な内容のループ計算を所定回数行う(S512)。ここで所定回数とは、CPU18がS512の処理に要する時間が、コプロセッサ32がS510の処理に要する最大の時間と等しく又はそれ以上となるために十分な回数をいう。S512のループ計算を終了すると、CPU18はカウンタ*i*を1だけデクリメントし、その後S508からS514の処理を繰り返す。S508からS514までの処理の繰り返しは、S508において「 $i > 0$ 」の条件が満たされなくなるまで、すなわち、秘密鍵Yを構成する全てのビットについてS510の処理が行われるまで継続される。一方、S508において「 $i > 0$ 」の条件が満たされなくなった場合は、図3のS318、S320と同じ処理が実行され(S516、S518)、一連の復号化処理が終了される。

【0025】(第3実施形態)次に、本発明に係る第3実施形態について説明する。図6は、本実施形態のICカード40の構成を示す図である。ICカード40は、タイマーモジュール20もコプロセッサ32も有さない点において第1実施形態のICカード10、及び第2実施形態のICカード30と異なるものである。

【0026】図7は、RSA_CALCコマンドを実行するときのICカード40の動作を示す流れ図である。RSA_CALCコマンドを実行する場合のICカード40の動作は、S702からS708までは図3に示したICカード10の動作S302からS708までの動

作と同一である。S708においてカウンタ*i*が「 $i > 0$ 」の条件を満たすと、CPU18は、図10に示したMontgomery積の計算を実行する(S710)。Montgomery積の計算が終了すると、CPU18は乱数を発生させ(S712)、取得した乱数に対応する回数だけ所定のループ計算を実行する(S714)。所定のループ計算とは、図5のS512におけるループ計算と同様に、復号化処理とは無関係な内容の計算である。

【0027】S714のループ計算を終了すると、CPU18はカウンタ*i*を1だけデクリメントし、その後、秘密鍵Yを構成する全てのビットについてMontgomery積の計算が行われるまでS708からS716までの処理を継続する。さらにS708からS714までの処理が全て終了すると、図3のS318、S320と同じ処理が実行され(S718、S720)、一連の復号化処理が終了される。

【0028】以上説明したように、本実施形態では、S710におけるMontgomery積の計算の後にループ計算を実行する。このために復号化処理の終了時、及びS720におけるレスポンスの送信時は、ループ計算を行った時間分だけ遅延する。しかも、ループ計算を実行する回数は乱数により無作為に決定されるために、遅延する時間の長さは不確定である。したがって、本実施形態では、レスポンスが送信される時と、S710において行われるMontgomery積の計算時間との間に相関がなく、第三者がレスポンスが送信される時を観測することにより、秘密鍵Yのビット構成を予測することが不可能となっている。

【0029】(その他の実施形態)なお、本発明は、上記実施形態に限定されるものではない。上記実施形態は、例示であり、本発明の特許請求の範囲に記載された技術的思想と実質的に同一な構成を有し、同様な作用効果を奏するものは、いかなるものであっても本発明の技術的範囲に包含される。

【0030】例えば、上記実施形態においては、ICカードに外部から暗号文を与え、これを復号化させる場合を例に説明をしたが、これは、ICカードに外部から平文を与え、その平文を暗号化させることであってもよい。また、上記実施形態では、RSA暗号を使用するICカードについて説明したが、これは、本発明の技術的範囲をなんら限定する意味のものではない。本実施形態の技術的思想は、鍵情報を用いて暗号化又は復号化処理を行うICカードであって、その暗号化又は復号化処理に要する時間が鍵情報のビット構成に依存するものに広く適用可能である。さらに、上記実施形態では、復号化処理の実行中又は実行後に復号化処理の終了時及びレスポンス信号の送信時を遅延させるための処理を行っているが、この種の遅延処理は、復号化処理の実行前に実行することであってもよい。

【0031】

【発明の効果】以上詳しく説明したように、本発明によれば、応答情報の送信時は、鍵情報の内容と相関関係を有さないで、第三者により鍵情報の内容が推測され、ICカードのセキュリティが害される恐れがない。

【図面の簡単な説明】

【図1】本発明の第1実施形態であるICカード10の構成を示す図である。

【図2】RSA_CALCコマンドのフォーマットを示す図である。

【図3】RSA_CALCコマンドを実行するときのICカード10の動作を示す流れ図である。

【図4】本発明の第2実施形態であるICカード30の構成を示す図である。

【図5】RSA_CALCコマンドを実行するときのICカード30の動作を示す流れ図である。

【図6】本発明の第3実施形態であるICカード40の

構成を示す図である。

【図7】RSA_CALCコマンドを実行するときのICカード40の動作を示す流れ図である。

【図8】リーダ・ライタ等の外部機器からICカードに情報を伝達する時の様子を示す説明図である。

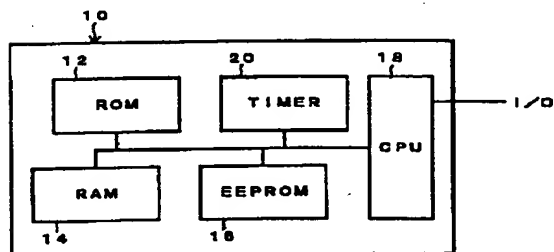
【図9】Montgomery法によるべき乗剰余の計算のアルゴリズムを示す図である。

【図10】Montgomery積の処理内容を示す流れ図である。

【符号の説明】

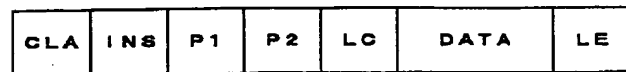
- 10 ICカード
- 12 ROM
- 14 RAM
- 16 EEPROM
- 18 CPU
- 20 タイマーモジュール
- 32 コプロセッサ

【図1】

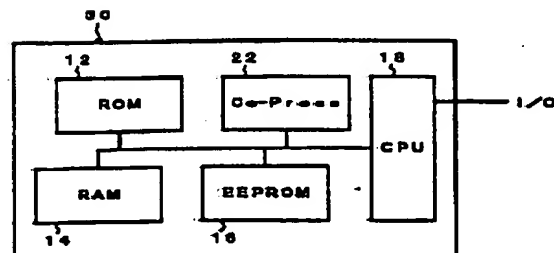


【図2】

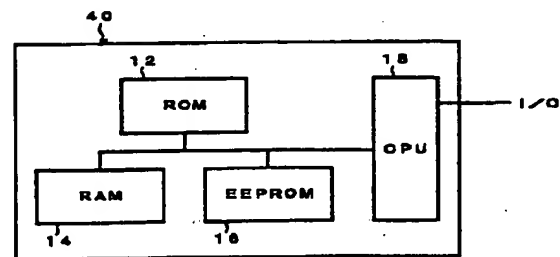
RSA_CALCコマンドのフォーマット



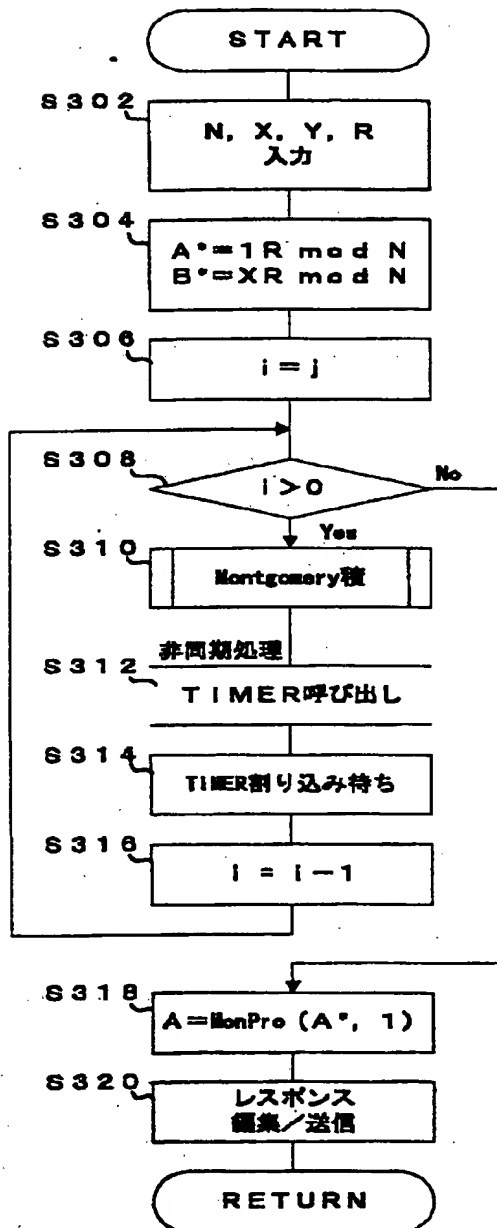
【図4】



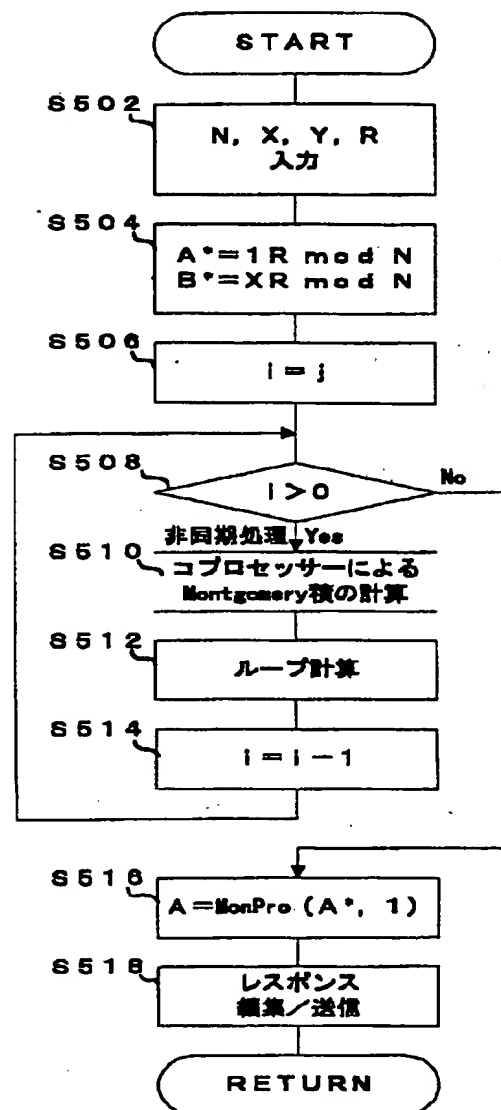
【図6】



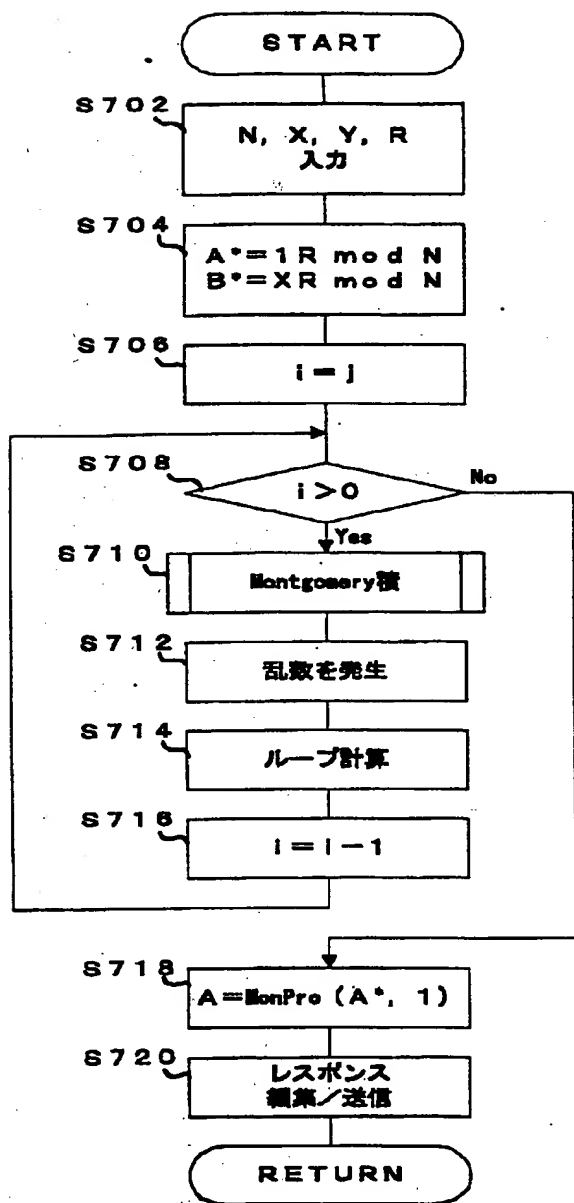
【図3】



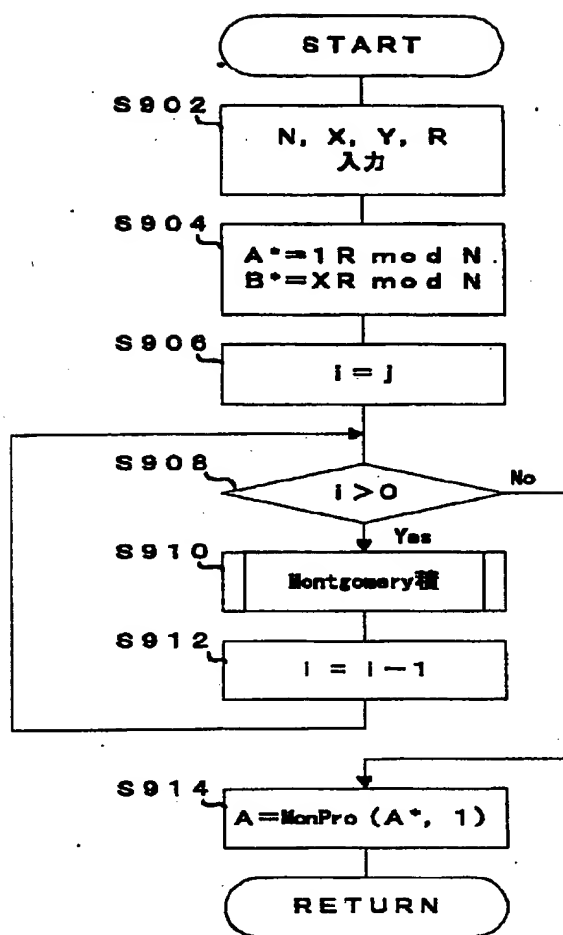
【図5】



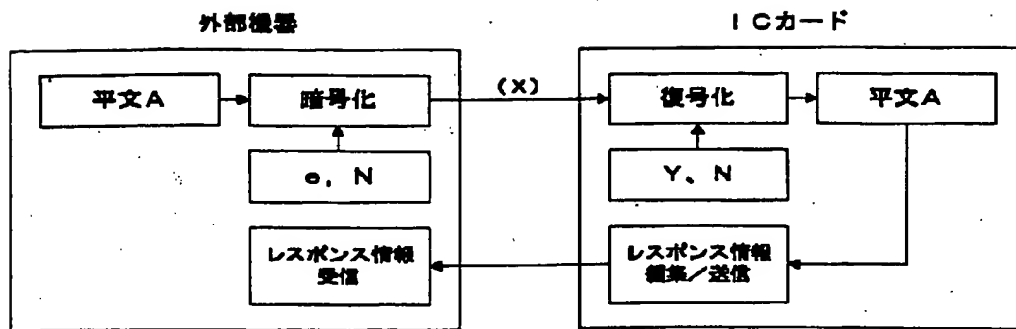
【図7】



【図9】



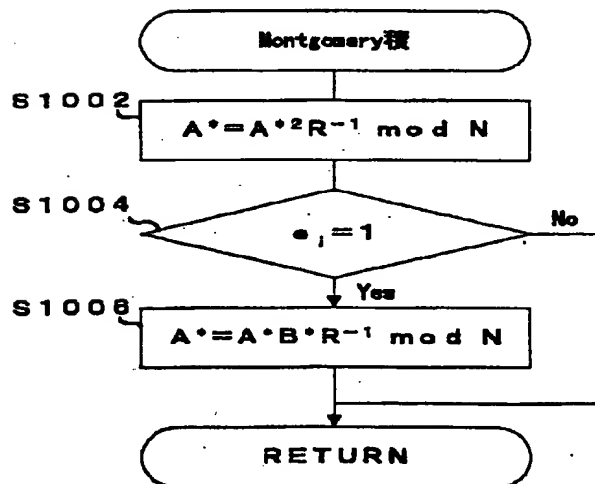
【図8】

暗号化: $X = A^e \bmod N$ 復号化: $A = X^Y \bmod N$

A: 平文
 e, N: 公開キー
 X: 暗号文

Y: 秘密キー

【図10】



フロントページの続き

(51)Int. Cl.⁸

H04L 9/08

9/10

9/30

識別記号

庁内整理番号

FI

H04L 9/00

技術表示箇所

601Z

621A

663A

JAPAN PATENT OFFICE

PATENT LAID-OPEN OFFICIAL GAZETTE

Laid-Open No.

H.10-69222

Laid-Open

H.10 (1998) Mar. 10

Application No.: H.8-224971

Filed: H.8 (1996) Aug. 27

Inventor: Kazuyoshi, Irisawa
1-1-1, Kaga-cho
Shinjuku-ku, Tokyo
Dainippon Printing Co., Ltd.

Tokio, Handa
1-1-1, Kaga-cho
Shinjuku-ku, Tokyo
Dainippon Printing Co., Ltd.

Applicant: 000002897
Dainippon Printing Co., Ltd.
1-1-1, Kaga-cho
Shinjuku-ku, Tokyo

Attorney, Agent: Hisao, Kamata

TITLE OF THE INVENTION

IC Card

ABSTRACT

[Problems]

To provide an IC card in which information about the key used for encrypting or decrypting cannot be estimated from the time at which the response to a command is transmitted.

[Means for solving the problem]

An IC card comprising a CPU and a memory which is accessible by the CPU, wherein the data is encrypted or decrypted using a key, in the form of information that has been stored in the memory, in response to an external command. The response to that result is transmitted beyond the card, and a delay process is applied, during or before and after encrypting or decrypting, to delay the time at which the response to a command is transmitted (S314 to S316). This is done so that the relationship between the time at which the response to a command is transmitted and information about the key will be removed.

WHAT IS CLAIMED

[Claim 1]

An IC card comprising a CPU and a memory which is accessible by said CPU, wherein data is encrypted or decrypted, on receipt of an external command, by using a key in the form of information stored in said memory, and the resulting information is transmitted outside the chip as the result, characterized in that:

there is a delay unit to delay the time at which the response to a command is transmitted; and

the relationship between the content of said key and the time at which the response to a command is transmitted

is removed by applying said delay unit during, or before and after, said encrypting or decrypting process.

[Claim 2]

An IC card, as described in Claim 1, characterized in that:

the operation of said delay unit is essentially separate from said encrypting and decrypting processes, and the CPU makes it take a random period of time.

[Claim 3]

An IC card as described in Claim 1, characterized in that:

it has a means for counting time which transmits an indication of the passage of a predetermined period; and

said delay unit is to suspend the start or continuation of said encrypting or decrypting by said CPU until said indication is transmitted from said means for counting time.

[Claim 4]

An IC card, as described in any of Claims 1 to 3, characterized in that:

said delay unit provides a fixed time for said encrypting or decrypting processes regardless of the bit configuration in said key.

DETAILED DESCRIPTION OF THE INVENTION

[0001]

[Scope of Utilization in Industry]

This invention concerns an IC card which encrypts or decrypts data in response to an external command.

[0002]

[Prior Art]

Recently, the focus has been on IC cards as a new information storage medium to replace magnetic cards. In particular, IC cards which incorporate CPUs are expected to find application in various fields in a highly information-oriented society because they can realize a high level of security since their information processing functions go beyond the function of being a medium for storing information. Generally speaking, an IC card incorporates a nonvolatile memory such as an EEPROM in which information is stored as files. An internal CPU accesses the EPROM on the basis of its interpretation and implementation of commands received from an external source. There are predetermined access conditions for each file. The CPU only accesses a file when its access conditions are satisfied by the arguments of the command. The IC card prevents illicit alteration or theft of data by third parties who do not have just title to the data.

[0003]

In addition, in an IC card system, data is encrypted when it is transferred between an IC card and a reader-writer. This prevents the content of a signal carrying information between the reader-writer and the IC card from being stolen, even when the signal is illicitly acquired by third parties. Figure 8 shows the transitions of information during transfer between external equipment such as reader-writers and an IC card. The external equipment produces cryptogram X by applying a predetermined encrypting to plain text A then transmits the cryptogram to the IC card. The IC card decrypts cryptogram X by decrypting which corresponds to the encrypting used by the external equipment. After the decrypting has been completed, the IC card edits information as a response to indicate the completion of this process then transmits this response to the external equipment. The given sequence of processing is completed when the external equipment receives this response.

[0004]

Plain text is usually encrypted by calculating the result of a formula with the plain text and a predetermined key as the variables. A typical example of a method for encrypting which is now proposed is the RSA cryptogram (refer to "A method for obtaining digital signatures and public-key cryptosystems" by L.Rivest, A.Shamir, and L.M.Adleman, *Communications of the ACM*, Vol. 21, No. 2: pp. 120—126, Feb. 1978). The RSA cryptogram is a so-called asymmetric key method of encryption, and this means that different key information is used for encrypting plain text and decrypting cryptograms. One of the two types of asymmetric key information is a secret key that is stored but kept secret from third parties and the other is used as a public key to be broadly made public to third parties.

[0005]

An RSA cryptogram is decrypted according to the formula " $A = X^Y \pmod{N}$ ". Here, A is the plain text, X is the cryptogram, Y is the secret key, and N is the public key. An RSA cryptogram is decrypted by calculating this modular exponential. Consequently, the amount of calculation required is usually enormous. Therefore, a calculation algorithm is used to reduce the calculation involved in obtaining the modular exponential. One known example of an algorithm that reduces the amount of calculation involved in this exponential function is the binary calculation of exponential functions (refer to *The Art of Computer Programming*, Vol. 2, Seminumerical Algorithms, by D.E.Knuth. Addison-Wesley, 2nd edition, 1981).

[0006]

In the calculation of an RSA cryptogram, this binary calculation of exponential functions is applied along with an algorithm called the Montgomery method, which allows modular operations to be carried out with relatively few

calculations (refer to "Modular Multiplication without Trial Division" by P.L.Montgomery, *Mathematics of Computation*, Vol. 44, No. 170, pp. 519—521, Apr. 1985). When the Montgomery method is applied, the modular calculations involved in calculating the modular exponential $A = X^Y \pmod{N}$ are carried out according to the algorithm shown in Figure 9.

[0007]

Firstly, the four values of N , X , Y , and R are input (S902). Here, X satisfies the relationship $0 < X < N$ with respect to N . In addition, secret key Y is expressed in binary form as $Y = e_j e_{j-1} \dots e_2 e_1$. Here, e_i is the value of the i th bit. R is defined as $R = 2^j$, using the value of bit j of Y . Next, Montgomery moduli A^* and B^* are determined for the input values (S904). The Montgomery modulus is defined with a one-to-one relationship with $(a \pmod{n})$, as $(a^* = ar \pmod{n})$, using n as a divisor. Here, n is a k -bit integer, $2^{k-1} \leq n < 2^k$, $r = 2^k$, and $\gcd(r, n) = 1$, where \gcd is the greatest common divisor.

[0008]

The Montgomery product is then calculated for each bit of exponent Y (S906 to S912). That is, when exponent Y is composed of j bits, the calculation in the Montgomery product step, S910, takes place j times. Here, the Montgomery product is a product, defined as $\text{MonPro}(a^*, b^*) = a^* b^* r^{-1} \pmod{n}$. Figure 10 is a flowchart of the processing involved in finding the Montgomery product. Two Montgomery product operations, S1002 and S1006, are involved in determining the Montgomery product. The operation S1002 is carried out regardless of whether the given bit of exponent Y is 1 or 0. On the other hand, the operation S1006 is only carried out for the respective bits of exponent Y that are 1. When the calculations that make up the loop (S906 to S912) have been completed for all bits of exponent Y , the Montgomery product of A^* and 1 (S914) is obtained. As a result, plain text A is acquired

by decrypting cryptogram X, and this completes the sequence of decrypting.

[0009]

[Problems to be Solved by the Invention]

As described above, the conventional decrypting process on an IC card differs according to whether or not each binary digit, as exponent Y is input, is 1. That is, in the processing shown in Figure 10, calculation takes place on the two steps S1002 and S1006 when the corresponding bit is 1 and on the single step S1002 when the corresponding bit is 0. Accordingly, the time required for decrypting depends on the number of 1-valued bits in the exponent, and increases with the number of 1-valued bits.

[0010]

Third parties are able to record the time elapsed between the transmission of the command for the decrypting of a cryptogram to the IC card and the sending back of a response, use this to find the time required for decrypting by the IC card, and then estimate the ratio of 1-valued and 0-valued bits in the index (secret key) Y used in the calculation of the modular exponential. That is, there is a problem with conventional IC cards in that the time when a response is transmitted may be used to crack the secret key Y and this is a danger to the security of such IC cards.

[0011]

So, the subject of this invention is the provision of IC cards in that prevent the use of the time taken for the response to an external command to be transmitted to estimate information about the key which was used for encrypting or decrypting.

[0012]

[Means for Solving the Problem]

In order to solve the above problem, the invention related to Claim 1 is an IC card comprising a CPU and a memory which is accessible by said CPU, wherein data is encrypted or decrypted, on reception of an external command, by using a key in the form of information stored in said memory, and the resulting information is transmitted outside the chip as the result, and characterized in that:

there is a delay unit to delay the time at which the response to a command is transmitted; and

the relationship between the content of said key and the time at which the response to a command is transmitted is removed by applying said delay unit during, or before and after, said encrypting or decrypting process.

[0013]

The invention related to Claim 2 is an IC card, as described in Claim 1, characterized in that:

the operation of said delay unit is essentially separate from said encrypting and decrypting processes, and the CPU makes it take a random period of time. Here, "the operation ... is essentially separate" means that normal encrypting or decrypting is possible, even if the unit does not operate. "take a random period of time" means, for example, that an operation which requires a fixed time for execution is executed a random number of times, or that an operation for which the execution time is randomly decided on each execution is executed once, twice, or more times.

[0014]

The invention related to Claim 3 is an IC card, as described in Claim 1, characterized in that:

it has a means for counting time which indicates the passage of a predetermined period; and

said delay unit is to suspend the start or continuation of said encrypting or decrypting by said CPU until said indication is transmitted from said means for counting time.

The invention related to Claim 4 is an IC card as described in any of Claims 1 to 3, characterized in that:

said delay unit provides a fixed time for said encrypting or decrypting processes regardless of the bit configuration in said key.

[0015]

[Embodiments of the invention]

This invention is described in detail below, on the basis of embodiments, with reference to drawings.

(First embodiment)

The first embodiment of this invention is an IC card that can use the RSA, an asymmetric-key encrypting method, to encrypt plain text or decrypt a cryptograph. In this embodiment, encrypting or decrypting by the RSA is applied as a numerical algorithm in which the Montgomery method is used, as shown in Figure 9. Figure 1 is a drawing of the configuration of an IC card related to this embodiment. As shown in Figure 1, IC card 10 is equipped with ROM 12 which is a read-only memory, RAM 14 which is a volatile memory, EEPROM 16 which is an electrically programmable nonvolatile memory, CPU 18 which has access to these memories, and timer module 20. Here, the timer module is a counting device, the operation of which is independent of that of CPU 18. It sends an interrupt to CPU 18 to indicate the passage of an assigned period of time.

[0016]

IC card 10 is equipped with an I/O line for the transfer of, for example, electric signals, and accompanied by a reader-writer (not shown). When an IC card is inserted in the reader-writer, a contact point is

connected with this I/O line, for the transfer of electric signals. Commands are issued to CPU 18 via said I/O line. A command is information sent to the IC card when the reader-writer requires a predetermined operation of the IC card. When a command is issued to CPU 18, a program stored in ROM 12 or EEPROM 16 is executed to process that command.

[0017]

Figure 2 shows the format of an RSA_CALC command, which is one of the commands used in this embodiment. The RAS_CALC command makes the IC card 10 decrypt a cryptogram and send back the plain text obtained thereby as a response. The first five bytes of the RSA_CALC command are respectively; CLA that indicates the class of a command, INS that indicates a classification, P1 and P2 that are the parameters of the command, and LC that indicates the length of subsequent DATA (as a number of bytes). The DATA after the sixth byte is the cryptogram X to be decrypted. Furthermore, the single LE byte that follows DATA is the expected length of the response. The value of LE is set so that all encrypted or decrypted data is thus sent back with a maximum length of 256 bytes in this embodiment.

[0018]

Figure 3 is a flowchart that indicates the operation of IC card 10 when the RSA_CALC command is executed. When the RSA_CALC command is received, cryptogram x is acquired from the DATA part of RSA_CALC, public key N, secret key Y, and constant R are then read from the predetermined addresses of the EEPROM (S302) in which they are stored. Next, Montgomery moduli A* and B* are calculated from said acquired values (S304). Furthermore, counter i is set to the number of digits (number of bits) j in binary-valued secret key Y. (S306). A judgement of whether counter i is greater than 0 is then made (S308).

[0019]

When counter i is determined to be greater than 0 by judgement step S308, bits for which the Montgomery product has not been calculated must remain among the bits that configure secret key Y . In this case, CPU 18 calculates the Montgomery product for the bit of the key which is currently indicated by counter i (S310). S310 represents the same process as described in S910 in Figure 9. CPU 18 specifies a predetermined time to timer module 20 and starts it up at the same time as the processing of S310 starts. Here, the predetermined time is a time which is equal to or greater than that required to calculate the Montgomery product in S310 for a 1-valued bit. Timer module 20 thus counts for a predetermined time in parallel with the processing by CPU 18 of S310.

[0020]

CPU 18 waits for the interrupt from timer module 20 after the processing of S310 has been completed (S314). After the interrupt is received, CPU 18 decrements counter i (S316) and then repeats the processes from S308 to S316. On the other hand, counter i not satisfying " $i > 0$ " in S308 means that the Montgomery product has been calculated for all bits which configure secret key Y . In this case, CPU 18 calculates the Montgomery product of A^* and 1 as indicated in S318 to acquire plain text A . Furthermore, CPU 18 then edits this into a response into the form that is normally sent as a response to the RSA_CALC command, and transmits this to the reader-writer (S320).

[0021]

As described above, the IC card of this embodiment counts a predetermined time by starting up the timer module at the same time as the Montgomery product is being calculated in S310, and the next process is not executed even after the Montgomery product has been calculated so as to decrypt the cryptogram. This procedure fixes the calculation time for decrypting, always becomes fixed and

makes it independent of the bit configuration of secret key Y. In this embodiment, the time taken between IC card 10 receiving the RSA_CALC command and sending back its response thus becomes fixed and independent of the content of the secret key. Therefore, an IC card with an extremely high level of security can be provided, in which the bit configuration of the secret key cannot be estimated from the time at which the response to a command is transmitted.

[0022]

(Second embodiment)

Next, the second embodiment of this invention is described. In the following description, the same symbol is used for those parts that function in the same manner as parts of the first embodiment, to avoid duplication of description. Figure 4 shows the configuration of IC card 30 of this embodiment. IC card 30 differs from IC card 10, the first embodiment, in that it does not have timer module 20, but has co-processor 32, and uses this co-processor 32 to calculate the Montgomery products. In the same way as IC card 10, IC card 30 also decrypts cryptogram X on receipt of an RSA_CAL command from the reader-writer, and, when it has finished, transmits a response to indicate that the command processing has been completed. Decrypting is carried out on the basis of the RSA cryptogram, in the same way as in the first embodiment, and the calculation of the RSA cryptogram is according to the algorithm in which the Montgomery method is used.

[0023]

Figure 5 is a flowchart of the operation of IC card 30 when the RSA_CALC command is executed. In Figure 5, S502 to S508 have the same content as S302 to S308 in Figure 3, and there is no operational difference between IC card 30 and IC card 10. When the condition $i > 0$ is satisfied by counter i in S508, CPU 18 hands over the values of A^* , B^* ,

N, and R and the values of the respective bits of secret key Y to co-processor 32. Co-processor 32 calculates the Montgomery product terms such as A^* , as shown in Figure 10. Using the co-processor to calculate the Montgomery product allows faster decrypting by this embodiment.

[0024]

On the other hand, CPU 18 handles a loop calculation, which is independent of decrypting, a predetermined number of times while co-processor 32 calculates the Montgomery product (S512). Here, the predetermined number of times means a number of times such that the time required for processing S512 by CPU 18 is equal to or greater than the maximum time required for processing of S510 by co-processor 32. After the loop calculation in S512 has been completed, CPU 18 decrements counter i by 1 and the processing of S508 to S514 is then repeated. The repetition of the processing of S508 to S514 continues until $i > 0$ is no longer satisfied in S508, i.e., until processing by S510 has been applied to all of the bits which make up secret key Y. On the other hand, when $i > 0$ is not satisfied in S508, the same processing as in S318 and S320 of Figure 3 is carried out (S516 and S518), and the sequence of decrypting is complete.

[0025]

(Embodiment 3)

The third embodiment of this invention is now described. Figure 6 shows the configuration of IC card 40 in this embodiment. IC card 40 differs from IC card 10 of the first embodiment and IC card 30 of the second embodiment in that it has neither timer module 20 nor co-processor 32.

[0026]

Figure 7 is a flowchart that shows the operation of IC card 40 when an RSA_CALC command is executed. The operation of IC card 40 from S702 to S708 is the same as

the operation of IC card 10 from S302 to S308 shown in Figure 3. When counter i satisfies $i > 0$, CPU 18 calculates the Montgomery product shown in Figure 10 (S710). When the calculation of the Montgomery product is completed, CPU 18 generates a random number (S712) to carry out a predetermined loop calculation (S714) for the number of times corresponding to the acquired random number. The content of the predetermined loop calculation has no relation with decrypting, in the same way as the loop calculation in S512 of Figure 5.

[0027]

When loop calculation step S714 has been completed, CPU 18 decrements counter 18 by 1, then continues with processing from S708 to S716 until the Montgomery product has been calculated for all bits of secret key Y. In addition, when all processing from S708 to S714 has been completed, the same processing as in S318 and S320 is carried out (S718 and S720), and the sequence of decrypting is complete.

[0028]

As described above, the loop calculation takes place after the calculation of the Montgomery product in S710. Therefore, the time spent on the loop calculation delays the completion of decrypting and transmission of the response in S720. Furthermore, the number of times the loop calculation takes place is randomly defined by a random number, so the length of the delay is undefined. Accordingly, in this embodiment, there is no relationship between the time at which the response is transmitted and the time required for calculation of the Montgomery products in S710, so it is impossible for third parties to gain knowledge about the bit configuration of secret key Y by observing the time when the response is transmitted.

[0029]

(Other embodiment)

This invention is not limited to the above embodiments. These embodiments are examples, and all embodiments which have configurations, operations, and effects that are effectively applications of the same technical ideas as described in the Claims of this invention can be included in the technical field of this invention.

[0030]

For example, in the above embodiments, the case in which a cryptogram is given to an IC card from some external unit for decrypting is described. However, this could also be a method handling the encrypting of plain text issued to the IC card from some external device. Furthermore, an IC card in which an RSA cryptogram is used is described in the above embodiment, but this is not intended to mean that the technical scope of this invention is limited in this way. The technical idea of this embodiment is an IC card which uses a key for encrypting or decrypting, which is generally applicable to cases in which the time required for the encrypting or decrypting of a cryptogram is dependent on the bit configuration of the key information. Furthermore, in the above embodiment, processing is carried out to delay, during or after the decrypting process, the time at which decrypting is completed and the time at which the response signal is transmitted. However, such a delaying process can be applied before the decrypting process.

[0031]

Advantages of the Invention

As described in detail above, the time at which the response to a command is transmitted has no relationship with information about the key, so there is no possibility that such information about the key can be estimated by third parties who can thus impair the security of an IC card.

BRIEF DESCRIPTION OF THE DRAWINGS

[Figure 1] Drawing showing the configuration of IC card 10 for the first embodiment of this invention.

[Figure 2] Drawing showing the format of the RSA_CALC command.

[Figure 3] Flowchart of the operation of IC card 10 when the RSA_CALC command is executed.

[Figure 4] Drawing of the configuration of IC card 30 for the second embodiment of this invention.

[Figure 5] Flowchart of the operation of IC card 30 when the RSA_CALC command is executed.

[Figure 6] Drawing of the configuration of IC card 40 for the third embodiment of this invention.

[Figure 7] Flowchart showing the operation of IC card 40 when the RSA_CALC command is executed.

[Figure 8] Drawing describing the manner in which information is transferred between external equipment such as a reader-writer and an IC card.

[Figure 9] Drawing showing an algorithm for the calculation of the modular exponential executed by using the Montgomery method.

[Figure 10] Flowchart showing the content of the process for finding the Montgomery product.

Descriptions of Symbols

10: IC card

12: ROM

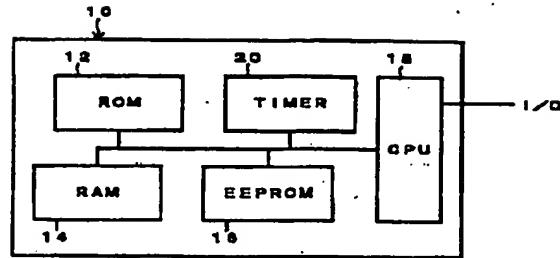
14: RAM

16: EEPROM

18: CPU

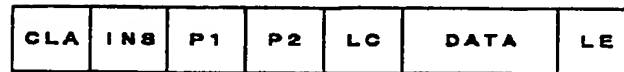
20: Timer module

32: Co-processor

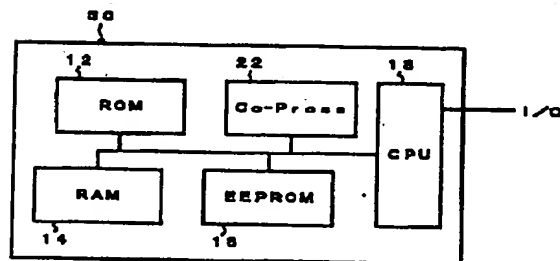


[Figure 1]

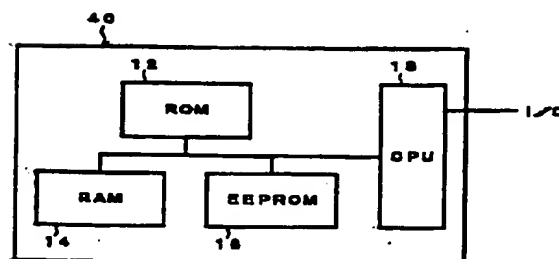
Format of the RSA CALC Command



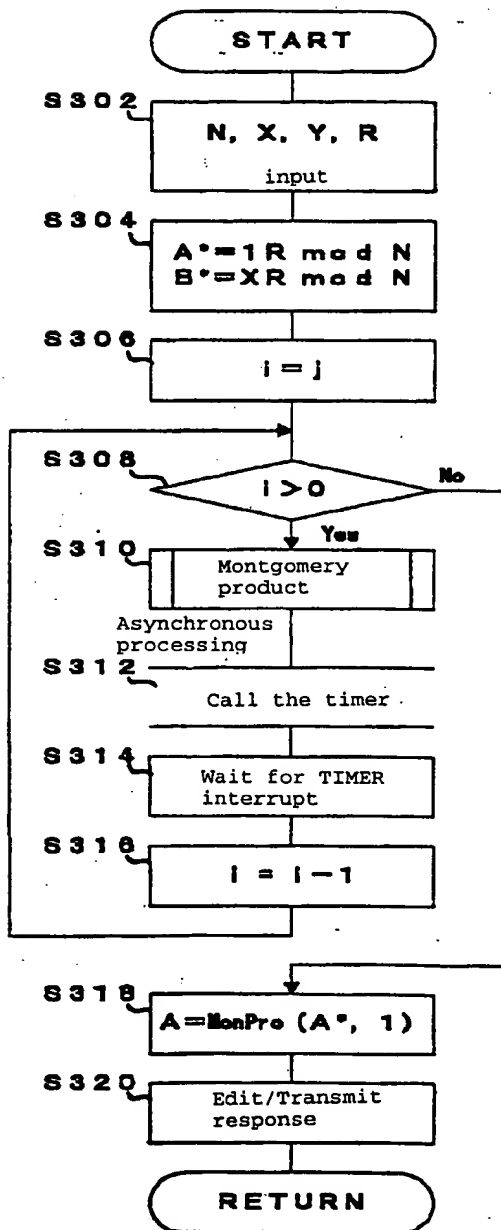
[Figure 2]



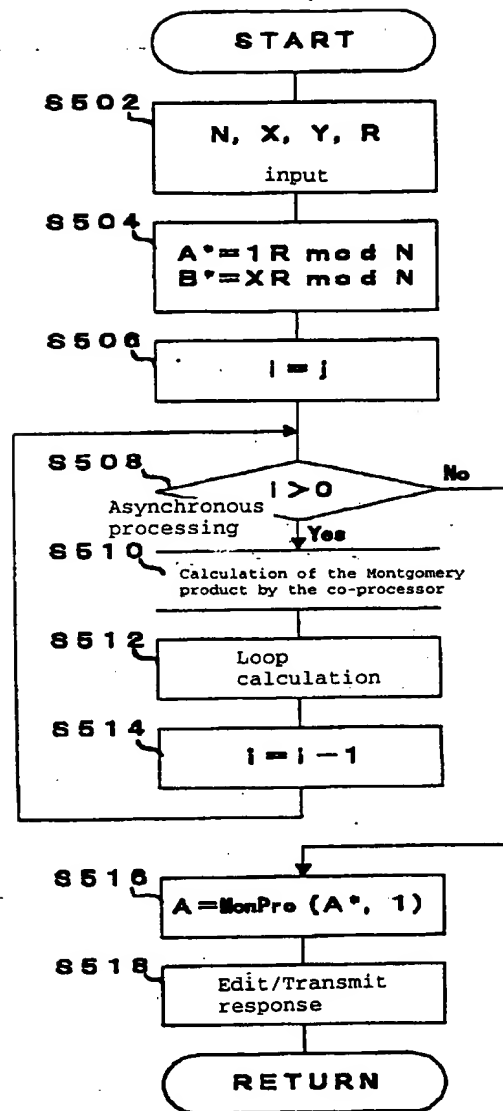
[Figure 4]



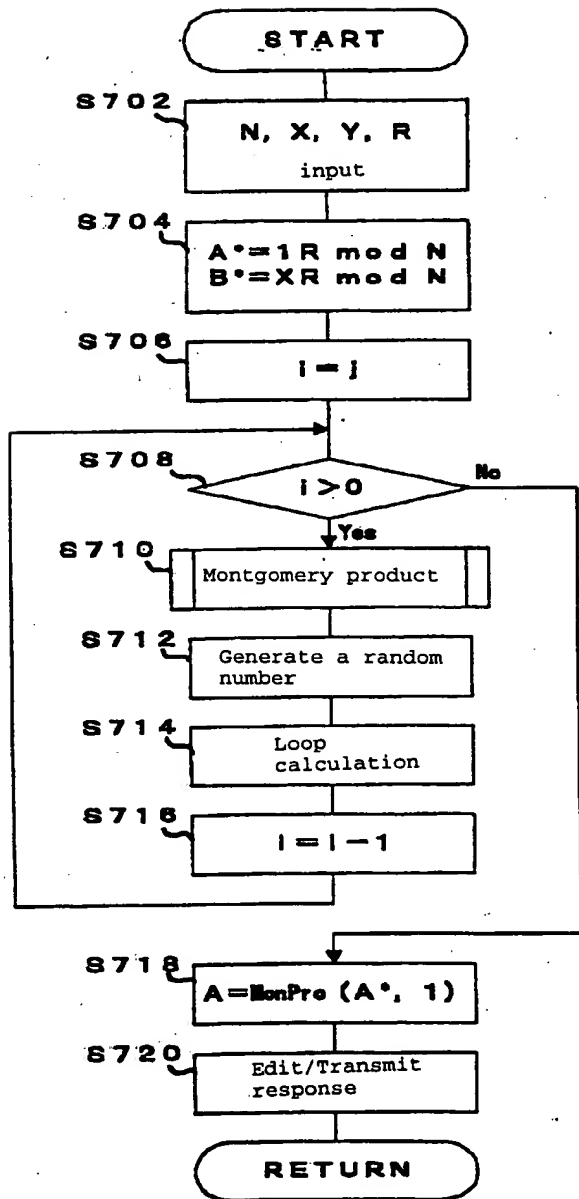
[Figure 6]



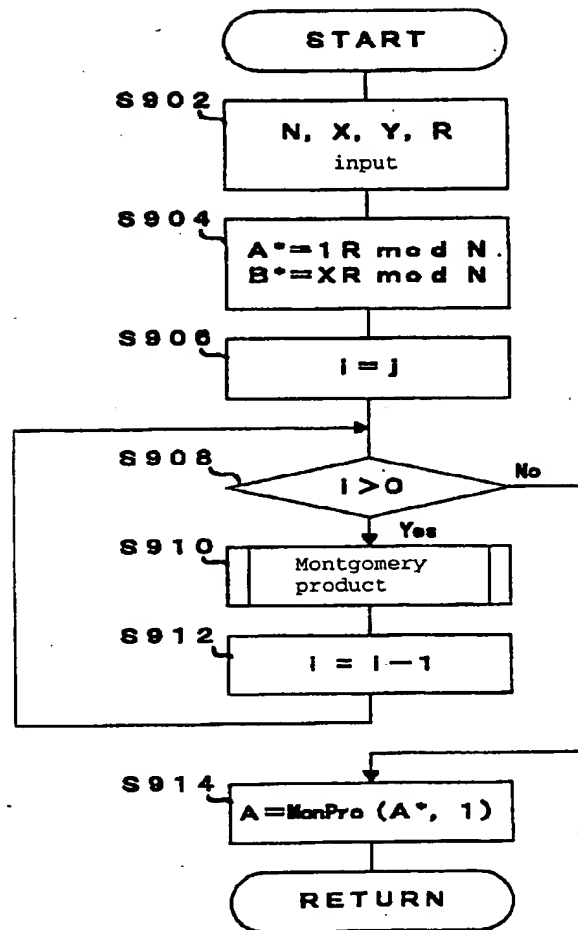
[Figure 3]



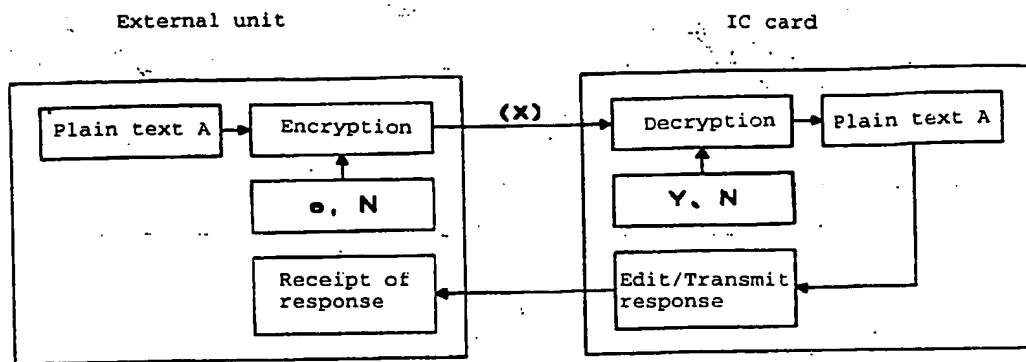
[Figure 5]



[Figure 7]



[Figure 9]

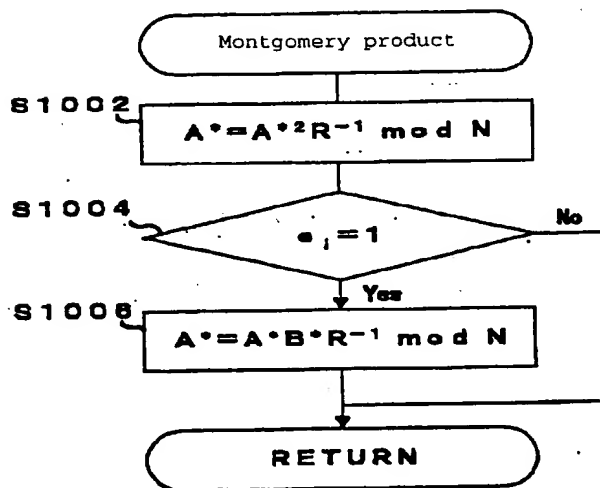


Encryption: $X = A^e \text{ mod } N$

Decryption: $A = X^Y \text{ mod } N$
Y: Secret key

A: Plain text
e, N: Public key
X: Cryptogram

[Figure 8]



[Figure 10]